HARDWARE/SOFTWARE LOCK FOR SECURE LINKAGE BETWEEN ALGORITHMIC DECISION PROCESS AND CRITICAL SYSTEM FUNCTION EXECUTION

Cross-Reference to Related Application(s)

This application is a continuation of U.S. Patent Application No. 09/802,321, filed on March 8, 2001, the specification of which is incorporated herein by reference.

Technical Field

The present invention relates generally to protection mechanisms for execution of system critical functions, and more particularly protection mechanisms for execution of system critical functions in a cardiac rhythm management system.

Technical Background

Given the broad spectrum of usage of microcomputers in controlling subsystems of mission critical systems, it is of utmost importance that the interface between the microcomputers and system elements implementing critical functions remain safe during fault conditions within the microcomputer. Such fault conditions may occur in either hardware or software part of the microcomputers. One such example is an unintended execution of otherwise normal software code due to incorrect state information such as program counter value, corrupt stack pointer or stack contents.

One class of mission critical systems is implantable cardiac rhythm management systems, in which electrical therapy is delivered to the heart. The same therapy sequences that are life sustaining may be life threatening if delivered with inappropriate timing. The decision to deliver electrical therapy generally involves the execution of complex algorithms that must be correctly executed to insure the appropriateness of delivered therapy. In the event of fault conditions that compromise the ability of the system to correctly execute such an algorithm, it is critical that

inappropriate therapy be blocked, and that the system operation be restored as quickly as possible.

Generally hardware and software locks are used to protect such system critical functions from inadvertent actuation. It is generally desired that such protection mechanisms be effective and efficient in requiring minimum hardware and software to provide satisfactory protection. Furthermore, it is desirable that the latency to actuation be well controlled and that the interval of time during which a critical task is accessible be short and well defined. Protection mechanisms that are effective in preventing unintended unlocking of actuation due to random processor actions are straightforward. They are generally a unique, necessary, and sufficient sequence of a software code (interacting with hardware), which is sufficiently unlikely to have been replicated either by localized or by non-localized random processor action.

Generally the method employed by mechanisms protecting against inappropriate therapy activation involves providing some degree of complexity in the interface to the hardware that activates the therapy. This can be illustrated, for example, with an interface to a microcomputer, that requires more than one bit be set in the input/output space before activation of the critical therapy can occur. Requiring a specific order or time sequence for setting the bits further restricts possible inadvertent activation, as does placing the bits at different input/output locations. With an interface of sufficient complexity, it is extremely unlikely that random activations of the interface by a malfunctioning subsystem or subsystems will result in therapy delivery.

This protection mechanism can generally be enhanced by placing a time limit on unlocking an actuation of a particular critical task, such that the likelihood of random actuation of that task in combination with the specific code to unlock that task within the requisite time period is even more unlikely. It is important to note that there is a limit to the efficacy of a software code sequence in preventing unintended random unlocking of actuation. To illustrate, assume that the unlocking software sequence is sufficiently improbable that only one occurrence of it exists in a device memory, and that occurrence is an intended key that unlocks the critical task. Further assume that

this code fragment is protected from being "run over" by a runaway processor and/or by software reset opcodes that precede it in memory. Under these assumptions, the only single random event that can cause execution of the fragment is if the instruction pointer is somehow pointed exactly at unlocking. This is not altogether an improbable event though the probability of its occurring is generally on the order of 2^{-16} or greater. Generally, however, such hardware locks are fairly easy to circumvent and by themselves afford little to no protection against unintended execution. Therefore, there is a need for an improved mechanism to protect from an unintended execution of critical tasks.

Summary of the Invention

According to one aspect of the present subject matter, a request to start a task is received by a first process. Then the first process informs a second process of running an algorithm to verify the legitimacy of the received request, determining the actual need to start the task. Then the second process stores the information regarding the starting of the algorithm by the first process. Then the first process runs the algorithm to verify the legitimacy of the received request, and conveys an outcome of the verification to the second process. Then the second process enables the start of the task by the first process based on the outcome of the verification and a checking of the stored information. Then the first process starts the task. In one embodiment the first process sets an interprocess token upon receiving the request to start the task. In another embodiment the first process unlocks a hardware/software lock upon enabling by the second process to start the task.

These and other embodiments, aspects, advantages, and features of the present invention will be set forth in part in the description which follows, and in part will become apparent to those skilled in the art by reference to the following description of the invention and referenced drawings or by practice of the invention. The aspects, advantages, and features of the invention are realized and attained by means of the instrumentalities, procedures, and combinations particularly pointed out in the appended claims and their equivalents.

Brief Description of the Drawings

In the drawings, like numerals describe substantially similar components throughout the several views. Like numerals having different letter suffixes represent different instances of substantially similar components.

Figure 1 is schematic/block diagram illustrating generally one embodiment of portions of a cardiac rhythm management including a system to prevent inadvertent actuation of a task and an environment in which it is used.

Figure 2 is a schematic drawing illustrating generally one embodiment of portions of a cardiac rhythm management device coupled to the heart shown in Figure 1 by a right atrial electrode and a right ventricular electrode.

Figure 3 is a schematic/block diagram illustrating generally one embodiment of portions of a cardiac rhythm management system including a system to prevent inadvertent actuation of a task according to the teachings of the present invention.

Figure 4 is a flow diagram illustrating generally one embodiment of operation of the present subject matter.

Detailed Description

In the following detailed description, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that the embodiments may be combined, or that other embodiments may be utilized and that structural, logical and electrical changes may be made without departing from the spirit and scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims and their equivalents. In the drawings, like numerals describe substantially similar components throughout the several views. Like numerals having different letter suffixes represent different instances of substantially similar components.

In this document the term hardware token is understood as requiring a specific order or time sequence for setting bits in a memory location that further restricts possible inadvertent activation. The term interprocess token refers to one process informing another process of running an algorithm to verify a legitimacy of a request to start a task by setting a token. The term hardware/software lock refers to an interface between a microcomputer and system elements implementing critical decisions having an ability to correctly execute system critical functions. Also, the term task refers to tasks such as delivering a high-energy electrical therapy to a heart experiencing cardiac arrhythmias such as atrial flutter, atrial and/or ventricular cardioversion/defibrillation and/or pacing therapy to a heart. The terms task and critical task are used interchangeably throughout the document.

General System Overview

This document describes, among other things, an improved mechanism to protect from unintended execution of critical tasks. Other aspects of the invention will be apparent on reading the following detailed description of the invention and viewing the drawings that form a part thereof.

Referring now to Figure 1, there is shown one example embodiment of a schematic/block diagram illustrating portions of a cardiac rhythm management system 100 and an environment in which the present invention can be used. In Figure 1, system 100 includes an implantable cardiac rhythm management device 105, also referred to as an electronics unit, which is coupled by an intravascular endocardial lead 110, or other lead, to a heart 115 of a patient 120. System 100 also includes an external programmer 125 providing wireless communication with device 105 using a telemetry device 130, and a distal end 140, which is coupled to one or more portions of the heart 115.

Referring now to Figure 2, there is shown a schematic drawing illustrating generally one embodiment of portions of the cardiac rhythm management device 105 electrically-coupled with the heart 115 by right atrial lead 110A and right ventricular lead 110B. The heart shown in Figure 2, includes a right atrium 200A, a left atrium

200B, a right ventricle 205A, and a left ventricle 205B. In this embodiment, the lead 110A includes electrodes (electrical contacts) disposed in, around, or near a right atrium 200A of the heart 115, such as a ring electrode 225A and tip electrode 220A, for sensing signals and/or delivering therapy to the heart's right atrium 200A. Also in this embodiment, the lead 110B includes electrodes disposed in, around, or near a right ventricle 205A of the heart 115, such as a ring electrode 225B and tip electrode 220B, for sensing signals and/or delivering therapy to the heart's right ventricle 205A. Leads 110A and B optionally also include additional electrodes, such as for delivering atrial and/or ventricular cardioversion/defibrillation and/or pacing therapy to the heart 115. Device 105 includes components that are enclosed in a hermetically sealed can 250. Additional electrodes may be located on the can 250, or on an insulating header 255, or on other portions of device 105, for providing unipolar pacing and/or defibrillation energy in conjunction with the electrodes disposed in or around the heart 115.

Referring now to Figure 3, there is shown by way of example, but not by way of limitation, one embodiment of portions of the cardiac rhythm management device 105, which is coupled with the heart 115. Device 105 includes a power source 300, and a controller 325. The device 105 includes a sensing circuit 305, an analyzer 330, a therapy circuit 320.

Sensing circuit 305 is coupled by atrial lead 110A and ventricular lead 110B to the heart 115 for receiving, sensing, and/or detecting electrical heart signals. Such heart signals can include atrial activations (also referred to as depolarizations or P-waves) which correspond to ventricular contractions. Such heart signals can include coordinated and uncoordinated cardiac rhythms. Such signals provided to the analyzer 330 can indicate, among other things, the presence of a cardiac arrhythmia. In one embodiment, the signals indicate atrial fibrillation and atrial flutter. Controller 325 also controls the delivery of therapy provided by the therapy circuit 320 and/or other circuits, as discussed below.

Controller 325 includes various modules, which are implemented either in hardware or as one or more sequences of steps carried out on a microprocessor or

other controller. Such modules are illustrated separately for conceptual clarity; it is understood that the various modules of controller 325 need not be separately embodied, but may be combined and/or otherwise implemented, such as in software/firmware.

In general terms, the sensing circuit 305 senses electrical signal from a heart tissue in contact with a catheter lead 110A or 110B to which the sensing circuit 305 is coupled. A sensed cardiac signal from the sensing circuit 305 is then received and processed by the analyzer 330 for the presence of a cardiac arrhythmia such as atrial fibrillation, atrial flutter, and/or ventricular cardioversion. Then the analyzer issues a request to a first processor 335 to start delivering a high-energy electrical therapy to the heart 115 based on the outcome of the analysis. Upon receiving the request to start a task such as delivering the high-energy electrical therapy by the first processor 335, the first processor 335 sets an interprocess token 350.

The controller 325 further includes a second processor 340 coupled to the first processor 335. The second processor 340 receives a first message from the first processor 335 about running an algorithm to verify a legitimacy of the received request to start delivering the high-energy electrical therapy. Upon receiving the first message from the first processor 335, the second processor 340 stores the received first message. Then the first processor 335 starts running the algorithm to verify the legitimacy of the request to start delivering the high-energy electrical therapy and sends a second message comprising an outcome of the verification to the second processor 340. Then the second processor 340 enables the start of delivering the high-energy electrical therapy by the first processor 335 based on the outcome of the verification. The controller 325 further includes a hardware/software lock 355 coupled to the first processor 335. The first processor 335 unlocks the hardware/software lock 355 to start delivering the high-energy electrical therapy upon enabling by the second processor 340 and further based on the status of the setting of the interprocess token 350.

In some embodiments, the high-energy electrical therapy is pacing pulse electrical energy. The high-energy electrical therapy can be a defibrillation pulse

electrical energy. The first and second processors 335 and 340 can be microprocessors or microcomputers. In some embodiments, the device 105 further includes a timer 370 coupled to the first processor 335, to provide a predetermined variable delay in electrical energy delivered through the electrodes 110B. The term critical task can mean a system critical function such as delivering a high-energy electrical therapy to the heart 115. In some embodiments, the first processor 335 starts an error checking routine when the task is not enabled by the second processor 340. In some embodiments, the first processor clears the interprocess token 350 after starting the task such as delivering the high-energy electrical therapy to the heart 115. The controller 325 can further include a memory 380. The setting of the interprocess token 350 can include writing to one or more memory locations in the memory.

Referring now to Figure 4, there is shown one embodiment of a method 400 of preventing an inadvertent actuation of a task in the event of a system failure according to the teachings of the present subject matter. At 410, a first process receives a request to start a task. At 420, the first process sends a first message regarding running an algorithm to verify the legitimacy of the received request to start the task, where the first message is sent to a second process. At 430, the second process records the first message sent by the first process. At 440, the first process runs the algorithm to verify the legitimacy of the request to start the task. At 450, the second process sends a second message indicating the need to start the task to the first process based on the outcome of the verification by the first process. At 460, the second process enables the first process to start the task based on checking the receiving of the first message from the first process.

In some embodiments, the first process sets an interprocess token upon receiving a request to start the task. The setting of the interprocess token can include writing to one or more memory locations. The first message can include informing the second process of setting the interprocess token by the first process. The method 400 can further include recording the setting of the interprocess token by the second process upon receiving information of setting of the interprocess token by the first process. The algorithm can include software code to verify the legitimacy of the

request to start the task. The second message can include requesting permission to start the task based on the outcome of the verification. The second message can also include an attempt to start the task based on the outcome of the verification. The method 400 can require the second process to enable the start of the task by the first process upon receiving the second message, and further based on the outcome of the checking of the status of the interprocess token. The method 400, can include setting a hardware/software lock upon enabling by the second process. The setting of the hardware/software lock can include unlocking the hardware/software lock to start the task.

The method 400, can further include running an error checking routine and clearing the interprocess token when the second message by the first process to the second process to start the task is not legitimate. Further, the method 400 can include running the error checking routine and the clearing the interprocess token when the second process does not enable the start of the task by the first process upon checking the status of the interprocess token. Also, the method 400 can include running the error checking routine when the hardware/software lock cannot be unlocked because the interprocess token is not set. The error checking routine under the above mentioned circumstances can be run by the first process. The method 400 can include clearing the interprocess token after starting the task. The task can include a system critical function such as delivering a high-energy electrical therapy to a heart. The system can be a cardiac rhythm management system, such as the one described with reference to Figure 1.

It is to be understood that the above description is intended to be illustrative, and not restrictive. Many other embodiments will be apparent to those of skill in the art upon reviewing the above description. The scope of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

Conclusion

The above described system provides, among other things, an improved protection mechanism to protect from unintended execution of critical tasks such as inadvertent actuation of a critical task due to a system failure.